

## Data storage method for actuator-sensor interface slaves

**Patent number:** DE19734694  
**Publication date:** 1999-03-04  
**Inventor:** BIHL JOCHEN [DE]; WIEDEMANN BERNHARD [DE]  
**Applicant:** BIHL JOCHEN [DE];; WIEDEMANN BERNHARD [DE]  
**Classification:**  
**- International:** H04L12/403; G06F13/12  
**- european:** G05B19/042; H04L12/403  
**Application number:** DE19971034694 19970811  
**Priority number(s):** DE19971034694 19970811

### Abstract of DE19734694

The data storage method involves saving the data sets received from complex apparatuses (4) which are coupled to actuator-sensor interface slaves (3) within the slaves and not in the host computer (1). An electronically erasable programmable read-only memory, a flash read-only memory, a read-only memory, or a random-access memory is used.

---

Data supplied from the **esp@cenet** database - Worldwide

99P 2875



37

19 BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENT- UND  
MARKENAMT

12 Offenlegungsschrift  
10 DE 197 34 694 A 1

51 Int. Cl.<sup>6</sup>:  
H 04 L 12/403  
G 06 F 13/12

21 Aktenzeichen: 197 34 694.4  
22 Anmeldetag: 11. 8. 97  
43 Offenlegungstag: 4. 3. 99

DE 197 34 694 A 1

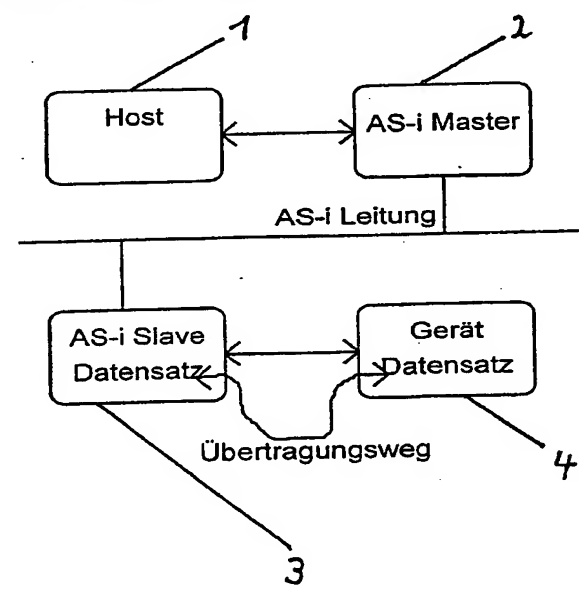
71 Anmelder:  
Bühl, Jochen, 68623 Lampertheim, DE; Wiedemann,  
Bernhard, 68161 Mannheim, DE

72 Erfinder:  
gleich Anmelder

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

54 Verfahren und Vorrichtung zur Speicherung von Datensätzen auf AS-Interface (AS-i) Slaves

57 Die Erfindung betrifft ein Verfahren, die Vorteile des AS-i wie geringe Kosten pro AS-i Slave (3), einfache Verdrahtung und eine ganz einfache Bedienung für die Anschaltung von komplexen Geräten (4) zu ermöglichen, wobei die Erfindung die Schwierigkeiten der Übertragung und Speicherung größerer Datenmengen über das AS-i kompensiert. Erfindungsgemäß werden Datensätze von an AS-i Slaves (3) angeschlossenen Geräten im AS-i Slave (3) und nicht im Host (1) gespeichert (s. Fig. 6), wobei der AS-i Slave (3) einen eigenen Datenspeicher (6) zum Beispiel ein EEPROM, ein FLASHROM, ein ROM oder ein RAM für das Speichern von gerätespezifischen Datensätzen verwendet und wodurch der Host (1) von der Speicherung der Datensätze entlastet wird und der Host (1) keinen Speicherplatz für die Datensätze mehr zur Verfügung stellen muß.



DE 197 34 694 A 1

## Beschreibung

## Technisches Gebiet

Die Erfindung betrifft eine Funktionserweiterung eines Bussystems zur Ansteuerung von miteinander vernetzten und kommunizierenden Slaves. Vorwiegend das Bussystem AS-Interface, weil dieses System aus Kostengründen möglichst einfach aufgebaut wurde, um die möglichst preisgünstige Vernetzung von einfachen binären Sensoren und Aktuatoren zu ermöglichen. Der preisgünstige Aufbau des Bussystems wurde unter anderem vor allen Dingen durch eine Beschränkung im Busprofil auf nur 4 Nutzbits erzielt.

Unabhängig von dieser Beschränkung ist AS-Interface ein Bussystem, bestens geeignet zur Vernetzung von Sensoren und Aktuatoren mit einem Hostrechner, wie z. B. speicherprogrammierbare Steuerungen oder Industrie-PC oder beliebige andere Busrechner. An den Hostrechner werden über einen AS-i Master die binären Sensoren/Aktuatoren angeschlossen, wobei der AS-i Master die auf das Bussystem von den AS-i Slaves aufgegebenen Signale in einem vorgegebenen Zeitraster (AS-i Masterprogramm) dem Hostrechner zur Verfügung stellt.

## Stand der Technik

Bekannte Feldbussysteme wie Profibus, Interbus u.s.w. waren für die Vernetzung einfachster Feldgeräte nur bedingt geeignet, da die Kosten für eine Anschaltung den Wert der einfachen Feldgeräte (Drucktaster, mechanische Endschalter, induktive Nähschalter etc.) weit übersteigt und auch die Vorteile der Verdrahtungsvereinfachung meist nicht ausreichen, um diese Mehrkosten auszugleichen. Das AS-Interface wurde geschaffen, um genau diese Lücke im Angebot der Feldbusse zu beseitigen. Die Anschaltkosten pro Slave erlauben erstmals die Integration der kompletten Slave-Elektronik in einfache und vor allem räumlich sehr kleine Sensoren und Aktuatoren. Dem Vorteil der preisgünstigen Slaveanschaltung steht der Nachteil der nur geringen Datenmenge pro vorgegebenem Zeitraster, durch die Beschränkung auf nur 4 Nutzbits zyklisch und 4 Parameterbits azyklisch gegenüber. Damit verwehrt sich der Einsatz dieses an und für sich idealen Bussystems für die Vernetzung komplexer Geräte, die zur Ansteuerung oder Parametrierung größere Datenmengen benötigen.

Die Erfindung beschreibt einen Weg diesen Nachteil zu vermeiden und auch über dieses einfach zu bedienende, in seiner topographischen Struktur einfach zu programmierende und vor allen Dingen sehr preiswerte Bussystem komplexe Geräte wie Antriebssteuerung (Frequenzumrichter) vernetzen zu können. Dabei bleiben alle Vorzüge des Bussystems erhalten.

Es sind zwar bereits Lösungen bekannt, wie über AS-i Bussysteme auch größere Datenmengen, z. B. 12 Bit Analogwerte übertragen werden können, auch sind bereits AS-i Slaves mit serieller Schnittstelle zum Feld hin bekannt. Diesen Lösungen ist aber gemeinsam, daß der Datensatz im Master aufgelöst werden muß und in einzelnen Datenpaketen zu je 4 Bit übertragen wird. Das belastet den Bus und benötigt Zeit, die in einer Steuerung für Maschinen und Anlagen meist nicht zur Verfügung steht.

Beschreibung des Anschlusses eines Slaves mit serieller Schnittstelle zum Feld hin mit den bekannten Methoden der Technik heute

Ein AS-i Slave überträgt maximal 4 Bit Eingangsdaten und 4 Bit Ausgangsdaten zyklisch und 4 Ausgabeparameterbits azyklisch. AS-i wurde zur zyklischen Übertragung von diesen binären Daten entwickelt. Es gibt darüber hinaus bereits AS-i Slaves, die größere Datenmengen, zum Beispiel einen 12 Bit Analogwert übertragen. Um diese größeren Datenmengen über AS-i zu übertragen wird ein Protokoll verwendet (z. B.: AS-i Analogprofil 7.1). Größere Datenmengen werden in Paketen mit jeweils 4 Bit mit einem überlagerten Protokoll übertragen.

Auch AS-i Slaves mit einer seriellen Schnittstelle sind verfügbar.

Diese ermöglichen den Anschluß von beliebigen Geräten mit serieller Schnittstelle an das AS-i. Die seriellen AS-i Slaves besitzen üblicherweise einen Mikrocontroller zur Bedienung der seriellen Schnittstelle.

Die seriellen AS-i Slaves arbeiten heute in zwei Betriebsmodi:

## 1. Zyklischer Betrieb

Im Zyklischen Betrieb ist ein spezielles Programm zur Bedienung von einem speziellen Endgerät im seriellen AS-i Slave hinterlegt. Den 4 Datenbits von AS-i sind feste Abläufe zugeordnet. Zum Beispiel bei einem Frequenzumrichter von Hersteller A. Ausgangsdatenbits:

Bit 0: Start/Stop, 1 bedeutet Start, 0 bedeutet Stop  
Bit 1: Rechts/Linkslauf, 1 bedeutet Rechtslauf, 0 bedeutet Linkslauf

Bit 2 und Bit 3: Auswahl von 4 Frequenzen

Bit 21 Bit 3

00: Frequenz 1 (niedrigste Frequenz)

01: Frequenz 2

10: Frequenz 3

11: Frequenz 4 (höchste Frequenz)

Eingangsdatenbits:

Bit 0: Start/Stop, 1 bedeutet Umrichter läuft, 0 bedeutet Umrichter ist gestoppt

Bit 1, Bit 2, Bit 3: Fehlermeldungen keine Kommunikation mit dem Umrichter Überstrom usw.

Vorteil gegenüber dem Anschluß an einen AS-i 4E/4A

Slave:

Größere Fehlersicherheit, es kann bei der seriellen Übertragung nicht mehr passieren, daß ein Ausgang gesetzt wird und diese Information unbemerkt (z. B.: durch einen Kabelbruch) nicht am Frequenzumrichter ankommt.

Darüber hinaus wird der serielle AS-i Slave mit weniger Leitungen angeschlossen.

Der Zyklischen Betrieb stellt keine besonderen Anforderungen an den Host.

## 2. Azyklischer Betrieb

Diese Betriebsart dient der zeitunkritischen Übertragung von Datensätzen. Im azyklischen Betrieb werden größere Datensätze in Paketen mit jeweils 4 Bit mit einem Protokoll übertragen. Upload ist das Lesen von Datensätzen aus einem Gerät. Download ist das Senden von Datensätzen an ein Gerät.

Upload und Download von größeren Datensätzen zwischen einem Host und einem Gerät ist in dieser Betriebsart über AS-i möglich. Der Übertragungsweg des Datensatzes ist dabei vom Host zum AS-i Master, vom AS-i Master zum AS-i Slave, vom AS-i Slave zum Endgerät und umgekehrt.

Die Geräte werden mit Hilfe spezieller Datensätze oft auf eine bestimmte Aufgabe abgestimmt. Zum Beispiel kann ein Frequenzumrichter mit einem Datensatz auf einen Motor oder auf eine spezielle Aufgabe abgestimmt werden.

Verschiedene Datensätze können im Host und im Gerät gespeichert werden.

Vorteil bei der Übertragung von größeren Datenmengen im azyklischen Betrieb: Mit Upload und Download können Datensätze zwischen dem Host und dem Gerät über AS-i übertragen werden. Nach dem Upload können Datensätze im Host gespeichert werden. Mit Hilfe des Download kann ein Gerät auf eine Betriebsart eingestellt werden. Angeschlossene Geräte können über AS-i spezielle Datensätze übertragen bekommen, beispielsweise nach dem Einschalten. Bei einem Gerätetausch im Fehlerfall wird das neue Gerät mit Hilfe der Datensätze wieder korrekt eingestellt. Darüber hinaus können Geräte mit verschiedenen Datensätzen über AS-i auf verschiedene Betriebsarten umgestellt werden.

Probleme bei der Übertragung von größeren Datenmengen im azyklischen Betrieb: Die Übertragung größerer Datenmengen in Paketen mit jeweils 4 Bit mit einem Protokoll ist für den angeschlossenen Host sehr aufwendig. Der Host muß genügend Rechenleistung zur Verfügung stellen. Das Protokoll muß für die verschiedenen am Markt befindlichen Hosts realisiert werden. In verschiedenen Hosts ist dies nicht möglich. Das Protokoll kann in verschiedenen Hosts nicht realisiert werden.

Die vielen am Markt befindlichen Endgeräte werden zu meist unterschiedlich bedient. Neben der aufwendigen Übertragung der Daten mit einem Protokoll über AS-i müssen in den verschiedenen Hosts auch spezielle auf die jeweiligen Geräte abgestimmte Regeln eingehalten werden. Ein Frequenzumrichter tauscht Daten mit anderen Mechanismen aus als ein induktives Identifikationssystem.

Darüber hinaus sind verschiedene Hosts, zum Beispiel Steuerungen, schlechte Speichermedien. Eine SPS oder Speicherprogrammierbare Steuerung hat üblicherweise keine Festplatte. Wenn beispielsweise ein Frequenzumrichter 240 Byte Daten in einem Datensatz besitzt, und es sind 31 Frequenzumrichter mit jeweils individuellen Parametersätzen angeschlossen, dann muß die SPS ca. 7,5 KByte speichern. Damit haben viele vor allem kleine Steuerungen große Schwierigkeiten.

Die Datensätze der Geräte werden üblicherweise mit Hilfe von Softwarepaketen oder sonstigen Werkzeugen der Gerätehersteller erzeugt. Die Datensätze können aus dem Gerät über die serielle Schnittstelle ausgelesen werden. Um den gerätespezifischen Datensatz in einem Host speichern zu können, muß der Datensatz über AS-i aus dem Gerät ausgelesen werden. Neben dem aufwendigen Senden von Datensätzen vom Host über AS-i zum Gerät muß auch der aufwendige Empfang von Datensätzen aus dem Gerät über AS-i realisiert werden. Dies erhöht den Aufwand erheblich.

Die Übertragungszeit von einem Datensatz über AS-i ist gegenüber der AS-i Zykluszeit von 5ms sehr lange. In einem AS-i Zyklus können systembedingt nur 4 Bit Daten übertragen werden. Ein Byte Daten besteht aus 8 Bit Daten. Ein Datensatz von 240 Byte könnte damit bestenfalls in 480 AS-i Zyklen oder in  $480 \cdot 5\text{ms} = 2,4$  Sekunden über AS-i übertragen werden. In der Realität ist die Übertragungszeit in Abhängigkeit des verwendeten Protokolls und der Rechengeschwindigkeit im Host deutlich länger.

Das Übertragen größerer Datenmengen über AS-i wird demzufolge heute sehr selten in der Industrie eingesetzt.

#### Technische Aufgabe

Der Erfindung liegt die Aufgabe zugrunde, ein Bussystem der eingangs genannten Gattung zu schaffen, bei dem die AS-i Slaves mit einer gewissen zusätzlichen Funktionalität ausgestattet sind, um Datensätze mit einem Gerät, mit weniger Aufwand für den Host auszutauschen, wobei der AS-i zur Steuerung oder Visualisierung eingesetzt werden kann.

#### Offenbarung der Erfindung und deren Vorteile

Die Erfindung betrifft ein Verfahren, die Vorteile des AS-i wie geringe Kosten pro AS-i Slave, einfache Verdrahtung und eine ganz einfache Bedienung für die Anschaltung von komplexen Geräte zu ermöglichen. Durch die Erfindung werden die Schwierigkeiten der Übertragung und Speicherung größerer Datenmengen über das AS-i kompensiert.

Die Lösung der Aufgabe besteht darin, erfindungsgemäß Datensätze im AS-i Slave zu speichern. Der AS-i Slave verwendet einen Datenspeicher zum Beispiel ein EEPROM, ein FLASHROM, ein ROM oder ein RAM für das Speichern von gerätespezifischen Datensätzen.

Bei einem Upload und Download von Datensätzen werden diese nicht mehr zwischen einem Host und dem Gerät, sondern zwischen dem AS-i Slave und dem Gerät ausgetauscht. Damit entfällt die von der Implementierung im Host aufwendige Übertragung der Datensätze zwischen dem AS-i Slave und dem Host. Die speziellen Regeln für die Bedienung eines Gerätes werden einmal im AS-i Slave implementiert und können von allen angeschlossenen Hosts ohne weiteren Entwicklungsaufwand genutzt werden. Die Übertragungszeit über AS-i entfällt ebenfalls.

Der angeschlossene Host wird entlastet. Der angeschlossene Host muß darüber hinaus keinen Speicherplatz für die Datensätze mehr zur Verfügung stellen.

Die Kontrolle, wann welcher Datensatz zwischen dem AS-i Slave und einem Gerät ausgetauscht wird, kann weiterhin durch den Host erfolgen, beispielsweise durch die Übertragung von AS-i Parametern und AS-i Daten. Die im AS-i Slave gespeicherten Datensätze werden durch die 4 Bit Eingangsdaten, 4 Bit Ausgangsdaten und oder die 4 Ausgabe-parameterbits angesprochen. Der Austausch von Datensätzen zwischen dem AS-i Slave und dem angeschlossenen Gerät wird in Abhängigkeit der Bits des AS-i Systems aktiviert. Damit bietet die Erfindung die Möglichkeit, mit den wenigen, in allen Hosts zur Verfügung stehenden Bits des AS-i Systems Geräte zu bedienen, die einen umfangreichen Datenaustausch zur Anpassung auf unterschiedliche Aufgaben benötigen. Hierbei wird weder der AS-i Bus noch der Host belastet. Dies ist sehr einfach mit fast jedem Host möglich. Die aufwendige Übertragung der Datensätze zwischen dem Host und dem Gerät über AS-i entfällt, da die Daten direkt im AS-i Slave gespeichert werden.

Es ist darüber hinaus die Bedienung direkt am AS-i Slave zum Beispiel über Taster und Anzeigen möglich. Upload und Download werden beispielsweise mit Hilfe von Tastern und Anzeigen direkt am AS-i Slave durchgeführt. Der Host wird in diesem Anwendungsfall für Upload und Download nicht benötigt. Damit ist auch dieser Betrieb mit jedem beliebigen Host möglich.

Kurzbeschreibung der Zeichnungen, in der zeigen:

Fig. 1 ein Beispiel für den Anschluß von Geräten über AS-i an einen Host,

Fig. 2: ein Beispiel für den Anschluß eines Endgerätes über den serieller AS-i Slave an AS-i,

Fig. 3 ein Blockschaltbild über den Aufbau eines seriellen AS-i Slave,

Fig. 4 Speicherort und Übertragungsweg von Datensätzen bei AS-i,

Fig. 5 ein Blockschaltbild eines Beispiels eines AS-i Slave mit Speicher gemäß der Erfindung,

Fig. 6 Speicherort und Übertragungsweg von Datensätzen gemäß der Erfindung,

Fig. 7 eine beispielhafte Bedienung der Datensätze gemäß der Erfindung durch den Host,

Fig. 8 eine beispielhafte Bedienung der Datensätze gemäß der Erfindung mit einem Taster und zwei LEDs.

## Bevorzugte Ausführung der Erfindung

In Fig. 1 ist ein Beispiel für den Anschluß von Geräten über AS-i an einen Host (1) dargestellt. Der Host (1) ist über den AS-i Master (2) und über die AS-i Slaves (3) mit den Geräten (4) verbunden.

In Fig. 2 ist ein Beispiel für den Anschluß eines Endgerätes (4) über den seriellen AS-i Slave (5) an AS-i dargestellt. Die serielle Verbindung zwischen dem Gerät (4) und dem seriellen AS-i Slave (5) kann beispielsweise eine RS232, eine RS485, eine RS422 oder eine Schnittstelle mit TTL (TTL: Transistor Transistor Logik) sein.

In Fig. 3 ist ein Blockschaltbild über den Aufbau eines seriellen AS-i Slave dargestellt. Der serielle AS-i Slave besteht unter anderem aus einem AS-i IC und einem Mikrocontroller.

In Fig. 4 ist der Speicherort und der Übertragungsweg von Datensätzen bei AS-i dargestellt. Nach heutigem Stand der Technik werden Datensätze eines Gerätes im Gerät selbst (4) und im Host (1) gespeichert. Der Übertragungsweg der Datensätze erfolgt über den AS-i Master (2) und den AS-i Slave (3).

In Fig. 5 ist das Blockschaltbild eines Beispiels eines AS-i Slave mit Speicher (6) gemäß der Erfindung dargestellt. Der AS-i Slave besteht unter anderem aus einem AS-i IC und einem Mikrocontroller und erfindungsgemäß einem Speicher (6) zur Aufnahme von Datensätzen.

In Fig. 6 ist der Speicherort und Übertragungsweg von Datensätzen (3, 4) gemäß der Erfindung dargestellt. Erfindungsgemäß werden Datensätze eines Gerätes im Gerät selbst (4) und im AS-i Slave (3) gespeichert. Der Übertragungsweg der Datensätze erfolgt erfindungsgemäß nur noch zwischen dem Gerät (4) und dem AS-i Slave (3).

In Fig. 7 ist eine beispielhafte Bedienung der Datensätze gemäß der Erfindung durch den Host über AS-i dargestellt. Der AS-i Slave arbeitet im Upload, Download Betrieb, sobald der Host über AS-i den AS-i Parameterwert 4 im AS-i Slave einstellt. Zur Kennzeichnung am AS-i Slave könnten zwei LEDs mit der Bedeutung Upload und Download angebracht sein.

Die Steuerung erfolgt mit Hilfe der 4 Datenbits.

Ausgangsdatenbit 0: Start, Wechsel von 0 auf 1 bedeutet Start (steigende Flanke)

Ausgangsdatenbit 1: Upload/Download, 1 bedeutet Upload, 0 bedeutet Download

Ausgangsdatenbit 2 und 3: Auswahl von 4 Datensätzen Bit 2/Bit 3

00: Datensatz 1

01: Datensatz 2

10: Datensatz 3

11: Datensatz 4

Eingangsdatenbits:

Eingangsdatenbit 0: Start/Stop, 1 bedeutet Aktion läuft, 0 bedeutet keine Aktion

Eingangsdatenbit 1: Fehlermeldungen, 1 bedeutet Fehler, 0 bedeutet keine Fehler.

In Fig. 8 ist eine beispielhafte Bedienung der Datensätze gemäß der Erfindung mit einem Taster und zwei LEDs dargestellt. Mit Hilfe eines Tasters und zwei LEDs (LED Upload und LED Download) kann ein Datensatz ausgetauscht werden. Sobald der Taster kurz betätigt wird, leuchtet die LED Upload auf und der Serielle AS-i Slave geht in den Handbedienmodus. Im Handbedienmodus wird das AS-i Slave ASIC im Reset gehalten. Es findet keine AS-i Kommunikation statt. Durch einen erneuten kurzen Tastendruck kann auf die LED Download umgeschaltet werden, und mit einem erneuten kurzen Tastendruck wieder zurück zu Upload usw.

Nachdem man die richtige Funktion ausgewählt hat, hält man den Taster 5 Sekunden gedrückt und versetzt den seriellen AS-i Slave in den jeweiligen Zustand. Danach beginnt die ausgewählte LED zu blinken. Die eigentliche Aktion, Upload oder Download wird durch einen weiteren kurzen Tastendruck ausgelöst. Solange die Funktion durchgeführt wird, flackert die jeweilige LED. Nachdem die Funktion fehlerfrei durchgeführt wurde, verläßt der serielle AS-i Slave den Handbedienmodus. Tritt ein Fehler auf, dann blinken beide LEDs 3 Sekunden abwechselnd und anschließend verläßt der serielle AS-i Slave den Handbedienmodus. Erfolgt 10 Sekunden keine Eingabe, so verläßt der Serielle AS-i Slave den Handbedienmodus. Damit kann der Handbedienmodus zu jedem Zeitpunkt abgebrochen werden, in dem man den Taster nicht mehr betätigt.

## Gewerbliche Anwendbarkeit

Der Gegenstand der Erfindung ist insbesondere für Bussysteme bei einem Aktuator-Sensor-Interface zur Steuerung und Aktivierung von miteinander vernetzten und kommunizierenden binären, Mehrfachmodulen, AS-i Slaves, anwendbar, deren Leistungsfähigkeit innerhalb des Aktuator-Sensor-Interfaces gemäß AS-i Standard wesentlich erhöht wird.

## Bezugszeichenliste

- 1 Host
- 2 AS-i Master
- 3 AS-i Slave
- 4 Gerät
- 5 serieller AS-i Slave
- 6 Speicher

## Patentansprüche

1. Verfahren, die Vorteile des AS-i wie geringe Kosten pro AS-i Slave (3), einfache Verdrahtung und eine ganz einfache Bedienung für die Anschaltung von komplexen Geräten (4) zu ermöglichen, wobei die Erfindung die Schwierigkeiten der Übertragung und Speicherung größerer Datenmengen über das AS-i kompensiert, dadurch gekennzeichnet, daß erfindungsgemäß Datensätze von an AS-i Slaves (3) angeschlossenen Geräten (4) im AS-i Slave (3) und nicht im Host (1) gespeichert werden, wobei der AS-i Slave (3) einen eigenen Datenspeicher zum Beispiel ein EEPROM, ein FLASHROM, ein ROM oder ein RAM für das Speichern von gerätespezifischen Datensätzen verwendet und wodurch der Host (1) von der Speicherung der Datensätze entlastet wird und der Host (1) keinen Speicherplatz für die Datensätze mehr zur Verfügung stellen muß.
2. Verfahren nach Anspruch 1 dadurch gekennzeichnet, daß erfindungsgemäß bei einem Upload und Download von Datensätzen diese nicht mehr zwischen einem Host (1) und dem Gerät (4), sondern nur noch zwischen dem AS-i Slave (3) und dem Gerät (4) ausgetauscht werden, wodurch die von der Implementierung im Host (1) aufwendige Übertragung der Datensätze zwischen dem AS-i Slave (3) und dem Host (1) entfällt und die speziellen Regeln für die Bedienung eines Gerätes (4) nur noch einmal im AS-i Slave (3) implementiert werden müssen, wobei die Datensätze von allen angeschlossenen Hosts (1) ohne weiteren Entwicklungsaufwand genutzt werden können und die Zeit für die Übertragung von Datensätzen über AS-i entfällt

3. Verfahren nach Anspruch 1 und 2 dadurch gekennzeichnet, daß die Kontrolle, wann welcher Datensatz zwischen dem AS-i Slave (3) und einem Gerät (4) ausgetauscht wird, weiterhin durch den Host (1) erfolgt, beispielsweise durch die Übertragung von AS-i Parametern und AS-i Daten und daß die im AS-i Slave (3) gespeicherten Datensätze nur durch die 4 Bit Eingangsdaten, 4 Bit Ausgangsdaten und oder die 4 Ausgabeparameterbits angesprochen werden und der Austausch von Datensätzen zwischen dem AS-i Slave (3) und dem angeschlossenen Gerät (4) in Abhängigkeit der Bits des AS-i Systems aktiviert wird, womit die Erfindung die Möglichkeit bietet, mit den wenigen, in allen Hosts (1) zur Verfügung stehenden Bits des AS-i Systems Geräte (4) zu bedienen, die einen umfangreichen Datenaustausch zur Anpassung auf unterschiedliche Aufgaben benötigen.

4. Verfahren nach Anspruch 1, 2 und 3 dadurch gekennzeichnet, daß die Bedienung der Datensätze direkt am AS-i Slave (3) zum Beispiel über Taster und Anzeigen ermöglicht wird, wodurch Upload und Download beispielsweise mit Hilfe von Tastern und Anzeigen direkt am AS-i Slave (3) bedient wird und der Host (1) in diesem Anwendungsfall für Upload und Download nicht benötigt wird.

---

Hierzu 5 Seite(n) Zeichnungen

---

30

35

40

45

50

55

60

65

Fig. 1 (Stand der Technik)

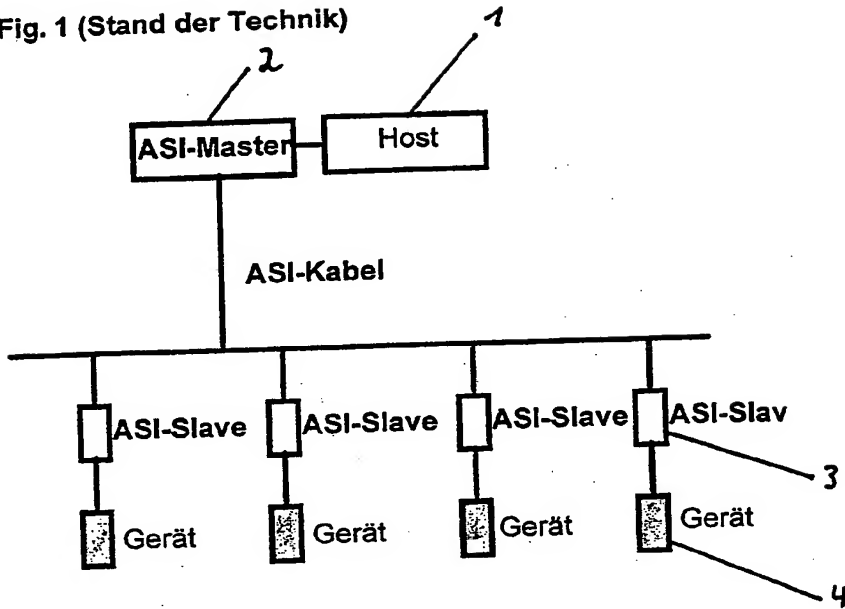


Fig. 2 (Stand der Technik)

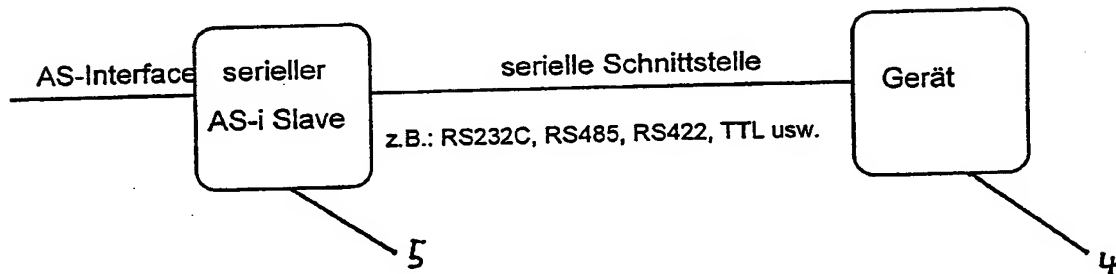


Fig. 3 (Stand der Technik)

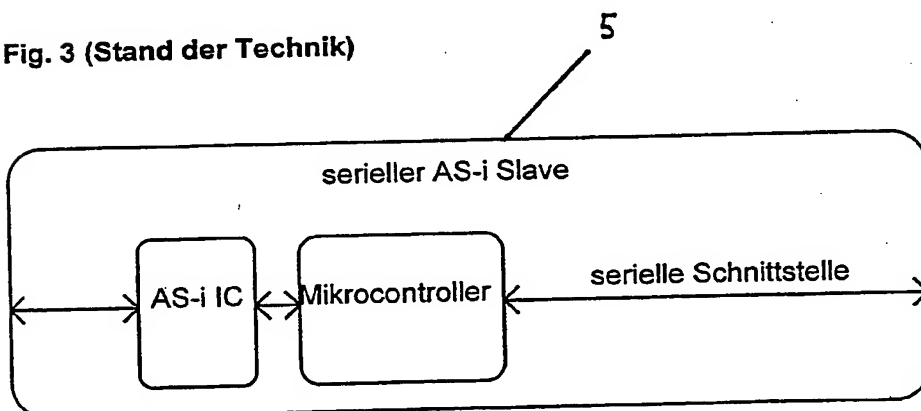


Fig. 4 (Stand der Technik)

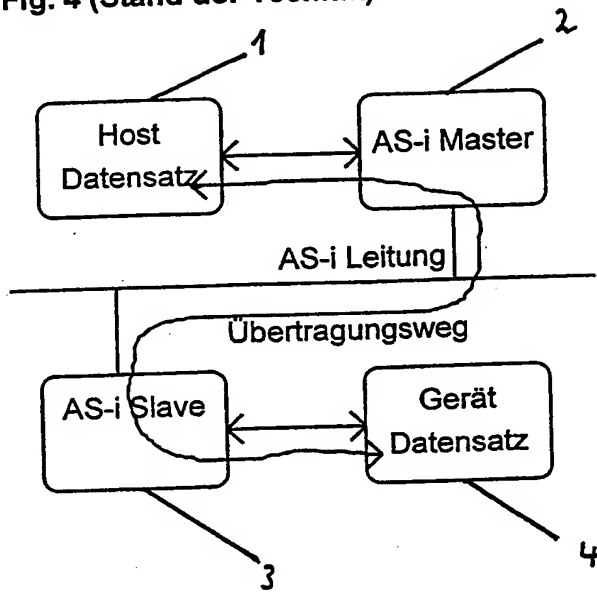


Fig. 5

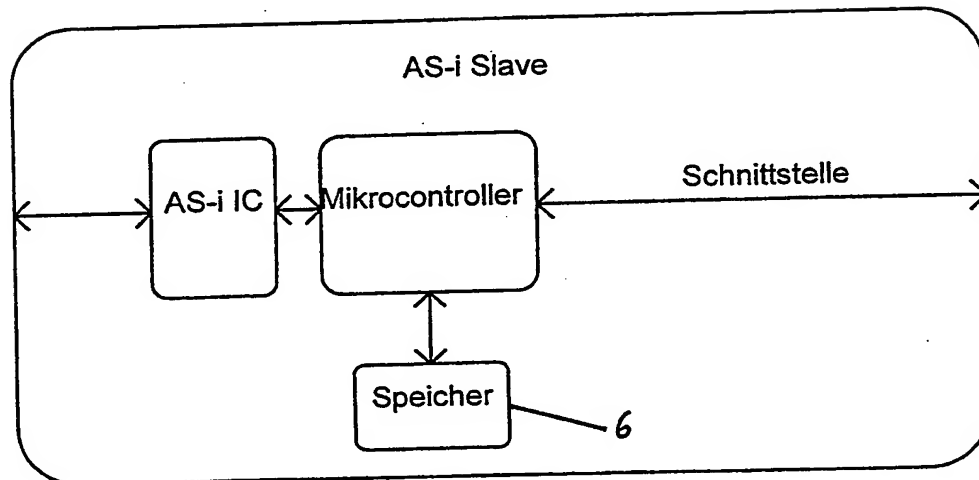




Fig. 6

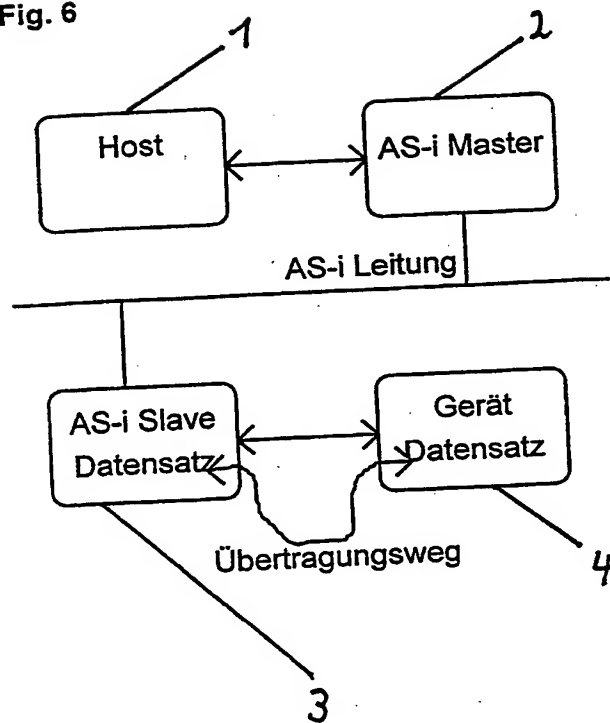


Fig. 7

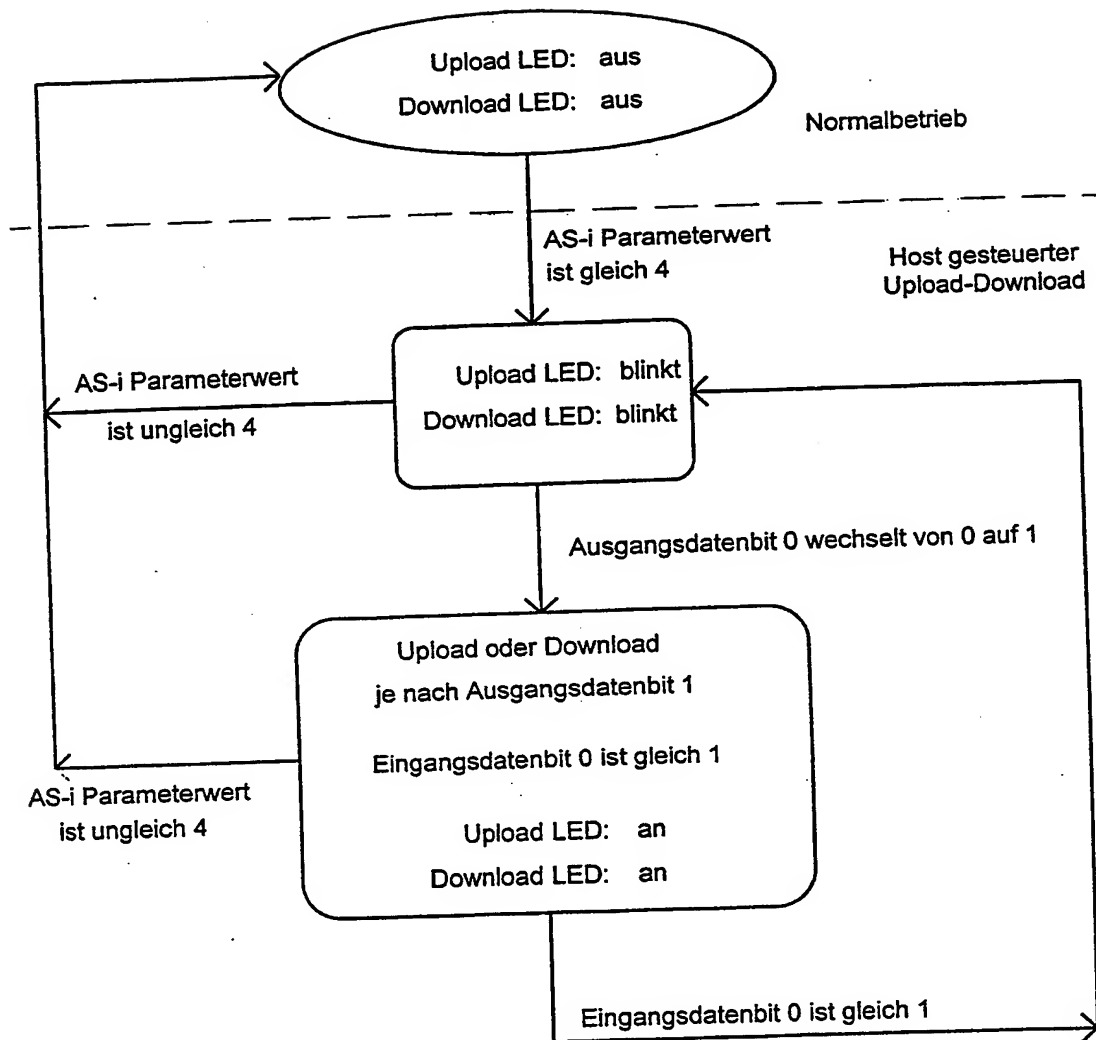
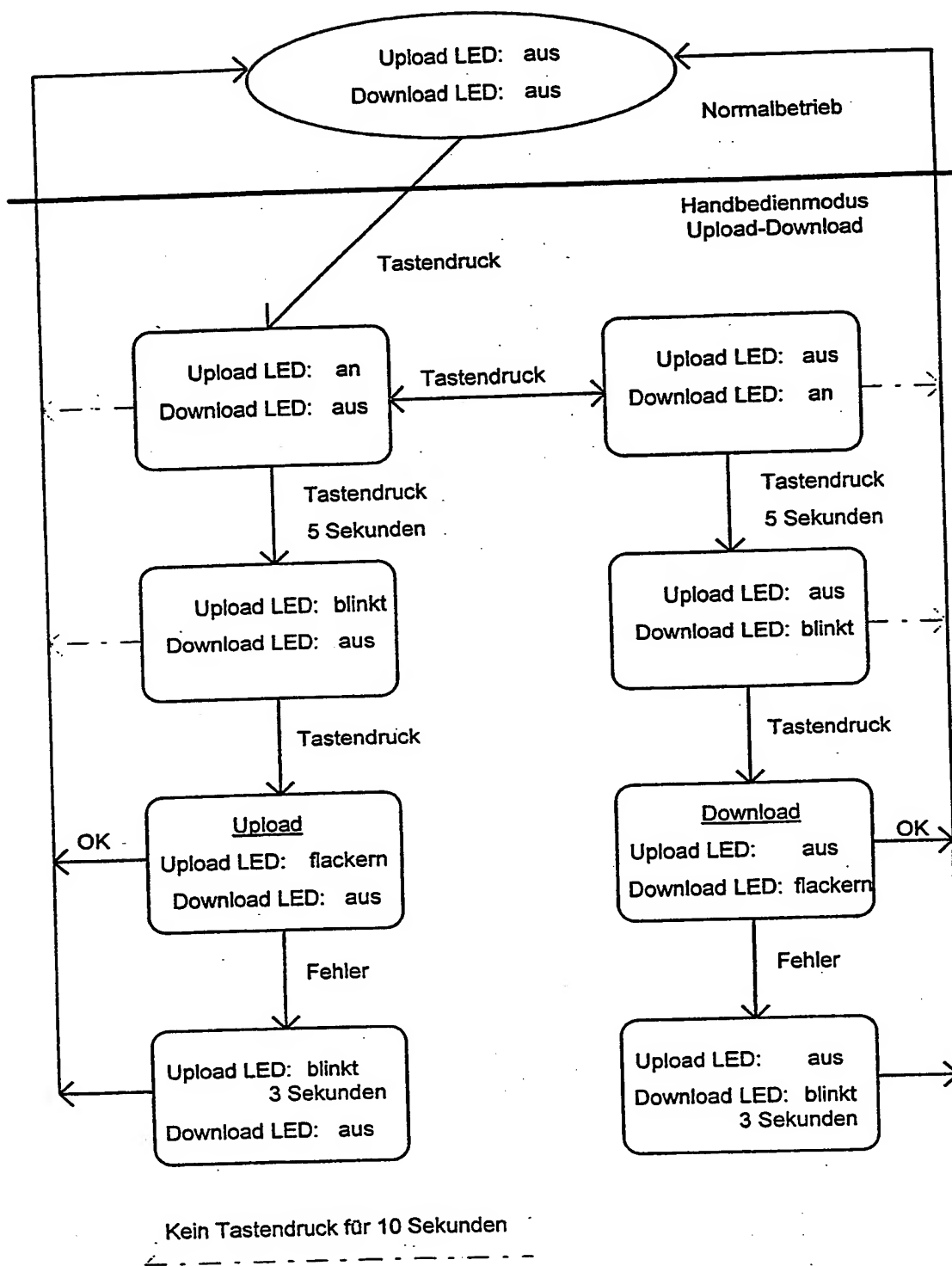


Fig. 8



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☒ **BLACK BORDERS**

☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

☐ **FADED TEXT OR DRAWING**

☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

☐ **SKEWED/SLANTED IMAGES**

☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

☐ **GRAY SCALE DOCUMENTS**

☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**

☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**